```python
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 18 12:19:57 2019

@author: Nate
"""

import numpy as np
import matplotlib.pyplot as plt
import useful as use
import pdb

#np.set_printoptions(threshold=np.nan)

sx = 6 #size of box
sy = 6
sz = 6

dt = .007 #timestep
t_total = 15000
#t_total = 1000


pause = 0.01

t_current = 0
N=27

x = np.array([1,3,5,1,3,5,1,3,5,1,3,5,1,3,5,1,3,5,1,3,5,1,3,5,1,3,5], dtype=float)
y = np.array([1,1,1,3,3,3,5,5,5,2,2,2,4,4.2,4,5.9,5.9,5.9,1,1,1,3,3,3,5,5,5], dtype=float)
z = np.array([1,1,1,1,1,1,1,1,1,3,3,3,3,3,3,3,3,3,5,5,5,5,5,5,5,5,5], dtype=float)
vx = np.array([0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0], dtype=float)
vy = np.array([0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0], dtype=float)
vz = np.array([0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0], dtype=float)
'''

x = np.array(np.random.uniform(0,sx,N),dtype=float)
y = np.array(np.random.uniform(0,sy,N),dtype=float)
vx = np.array(np.random.uniform(0,0,N),dtype=float)
vy = np.array(np.random.uniform(0,0,N),dtype=float)
z = np.array([0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0])
vz = np.array([0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0])
#np.zeros(20)
'''
kinetic = []
potential = []
total = []


pot = 0



##############################

#fig2, ax4 = plt.subplots()
#plt.title('Two Particles, Velocity')
```

```
#ax4.set_xlabel('Time-Step')
#ax4.set_ylabel('$V_x$ Coordinate')
#plt.xlim(0,sx)
#plt.ylim(0,sy)




def main(N,x,y,z,vx,vy,vz,sx,sy,sz,dt):

    global total, kinetic, potential, t_current

    ax = np.array([], dtype=float)
    ay = np.array([], dtype=float)
    az = np.array([], dtype=float)


    for i in range(N):
        ax = np.append(ax,0)
        ay = np.append(ay,0)
        az = np.append(az,0)


    ax, ay, az, pot = use.accel(N, x, y, z, ax, ay, az, sx, sy, sz)


    ek = 0.0
    for i in range(N):
        ek = ek + (vx[i]**2+vy[i]**2+vz[i]**2)
    kinetic.append(.5*ek)
    potential.append(pot)


    for i in range(t_total):
        #print(x)
        x,y,z,vx,vy,vz = use.update(N, x, y, z, vx, vy, vz, ax, ay, az, sx, sy, sz, dt, ek,


        t_current+=1
        p1 = [0,0]
        p2 = [0,0]
        #print(t_current)


        #if i % 100 == 0:

            #
            #ax4.scatter(x[0],y[0], marker='o', c = 'b')
            #ax4.scatter(x[1],y[1], marker='o', c = 'r')

         #   ax4.scatter(t_current,vx[0], c = 'b')
          #  ax4.scatter(t_current,vx[1], c = 'r')

            #p1 = ax4.scatter(x[0],y[0], marker='o', c = 'b', s = 150)
            #p2 = ax4.scatter(x[1],y[1], marker='o', c = 'r', s = 150)
            #p3 = ax4.scatter(x[2],y[2], marker='o', c = 'g')
            #p4 = ax4.scatter(x[3],y[3], marker='o', c = 'y')
```

```
            #plt.pause(pause)
            #p1.remove()
            #p2.remove()
            #p3.remove()
            #p4.remove()

    kinetic = np.asarray(kinetic)
    potential = np.asarray(potential)
    total = np.asarray(kinetic+potential)


    #plotting







main(N,x,y,z,vx,vy,vz,sx,sy,sz,dt)


'''
print('Initial Kinetic Energy: ',kinetic[:10], '\r\n\r\n')
print('Final Kinetic Energy: ',kinetic[-10:], '\r\n\r\n')
print('Initial Total Energy: ',total[:10], '\r\n\r\n')
print('Final Total Energy: ',total[-10:], '\r\n\r\n')
'''

avg_kinetic = np.asarray([])

for i in range(len(kinetic)):

    to_ap = 0
    for j in range(i):
        to_ap = to_ap + kinetic[j]
    to_ap = to_ap/(i+1)
    avg_kinetic = np.append(avg_kinetic, to_ap)

print(avg_kinetic[-5:])
fig1, ax3 = plt.subplots()
ax3.plot(kinetic)
ax3.plot(potential)
ax3.plot(total)
ax3.plot(avg_kinetic)
ax3.set_ylabel('Energy')
ax3.set_xlabel('Time-Step')
ax3.legend(('Kinetic','Potential','Total', 'Time Averaged Kinetic'), loc='upper right')
ax3.set_title("Energy VS Time", va='bottom')
plt.show()


#for i in range(25):
#    i = dt*i

#    use.update(np,x,y,z,vx,vy,vz,ax,ay,az,sx,sy,dt,pot,ek):
```